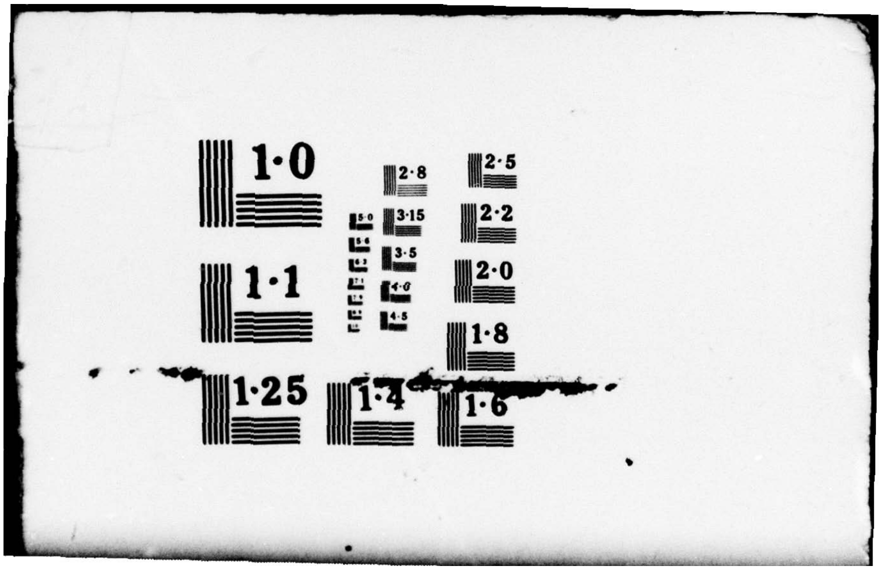
AD-A062 658   COLD REGIONS RESEARCH AND ENGINEERING LAB   HANOVER N H   F/G 13/2
COMPUTER FILE FOR EXISTING LAND APPLICATION OF WASTEWATER SYSTE--ETC(U)
NOV 78   I K ISKANDAR, D ROBINSON

UNCLASSIFIED   CRREL-SR-78-22   NL

1 OF 1
ADA
062658

END
DATE
FILMED
3-79
DDC


1·0
2·8
2·5
5·0
3·15
2·2
5·6
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

Special Report 78-22

November 1978

COMPUTER FILE FOR EXISTING LAND APPLICATION OF WASTEWATER SYSTEMS: A USER'S GUIDE

I.K. Iskandar, D. Robinson, W. Willcockson and E. Keefauver

Prepared for
DIRECTORATE OF CIVIL WORKS
OFFICE, CHIEF OF ENGINEERS
By
DEPARTMENT OF THE ARMY
COLD REGIONS RESEARCH AND ENGINEERING LABORATORY
CORPS OF ENGINEERS
HANOVER, NEW HAMPSHIRE

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
Special Report 78-22

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
6 COMPUTER FILE FOR EXISTING LAND APPLICATION OF WASTEWATER SYSTEMS: A USER'S GUIDE.

5. TYPE OF REPORT & PERIOD COVERED
9 Special rept.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
10 I.K. Iskandar, D. Robinson, W. Willcockson [redacted] E. Keefauver

8. CONTRACT OR GRANT NUMBER(s)
14 CRREL-SR-78-22

9. PERFORMING ORGANIZATION NAME AND ADDRESS
U.S. Army Cold Regions Research and Engineering Laboratory
Hanover, New Hampshire 03755

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Civil Works Project 31280

11. CONTROLLING OFFICE NAME AND ADDRESS
Directorate of Civil Works
Office, Chief of Engineers
Washington, D.C. 20314

12. REPORT DATE
11 Nov [redacted] 78

13. NUMBER OF PAGES
28

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
12 27p.

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer programs
Design data
Information retrieval
Land treatment
Planning
Wastewater treatment

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Two computer programs, both written in BASIC, have been developed to store and retrieve information on existing wastewater land treatment systems. The purpose of establishing these programs is to provide assistance to design engineers during the planning of new land treatment systems by making available the design criteria and performance characteristics of operating systems. The SEARCH program is designed to locate systems with specific design parameters, such as flow rate, waste type, application rate and mode, ground cover and length of operation. The printout from SEARCH includes a list of articles on similar systems in addition to the design parameters. The UPDATE program is used for the revision of information on file. Currently there are about 350 domestic and 75 foreign systems on file.

037 100

78 12 18 045

Preface

This report was prepared by Dr. I.K. Iskandar, Research Chemist, D. Robinson, Computer Technician, W. Willcockson, Environmental Science Technician, and E. Keefauver, Environmental Science Technician, of the Earth Sciences Branch, Research Division, U.S. Army Cold Regions Research and Engineering Laboratory.

The study was funded by Corps of Engineers Civil Works Project 31280, *Evaluation of Existing Facilities for Wastewater Land Treatment.*

The technical content of the report was reviewed by Thomas Jenkins and James Martell of CRREL.

SUMMARY

Land application of wastewater has been practiced for centuries all over the world. In recent years there has been a trend toward use of land application systems for the purpose of wastewater treatment. This is essentially a result of more stringent water quality standards and an increased concern over environmental pollution.

Since 1972 CRREL has conducted extensive research on land treatment of wastewater. A portion of this program has been concerned with evaluating the design, operating performance and management characteristics of existing land application systems and assessing the reasons for system success or failure.

The experience and information gained from existing systems, for example, information on the long-term environmental effects and problems encountered with heavy metals, should be utilized in the design and management of new systems. Use of such information may lead to a better means of dealing with such problems.

Due to the large number of existing systems and available publications and the diversity of sites, a computer file has been established at CRREL and made available to the public. The file contains compiled information on both foreign and domestic systems. Two programs have been written in BASIC to manipulate the file. The SEARCH program is used to locate a system or systems with specific design parameters and UPDATE is used to revise the file. A listing of both programs is included and a listing of the systems on file as of February 1978 is available in CRREL Internal Report 561. (Currently there are approximately 400 domestic and 75 foreign systems in the file.) SEARCH can be used along with the *Design Process Manual for Land Treatment of Wastewater*, published recently by the Corps of Engineers and Environmental Protection Agency, in planning and design of land application systems.

COMPUTER FILE FOR EXISTING LAND APPLICATION OF WASTEWATER SYSTEMS: A USER'S GUIDE

I.K. Iskandar, D. Robinson,
W. Willcockson and E. Keefauver

INTRODUCTION

Land application of wastewater is not new. The first documented wastewater and sludge applications on land were in Poland and Germany in the 16th century (Iskandar 1978, Wiezbicki 1977, Markland et al. 1974). In the United States, there have been systems in operation since the late 19th and early 20th centuries (Pound and Crites 1975). In recent years, wastewater application on land has been used for both the treatment of the wastewater and for water resources management (irrigation and recharging of underground water supplies). However, information on the performance and the long-term effects (if any) of older systems is needed for the design, operation and management of new systems.

Because of the large number of publications available and the diversity of sites and their environmental factors, a computer file has recently been established at CRREL in order to compile existing information and literature on both foreign and domestic systems. Information on these existing systems can be used in the design, operation and management of new land treatment systems. Also, an assessment of the long-term effects on the environment as a result of applying wastewater on land can be obtained. At the present time, information on more than 400 domestic and 75 foreign systems is stored in the computer. The purpose of this report is to describe to the user how to search for a system (or systems) and how to update systems on file. The two programs are named "SEARCH" and "UPDATE," respectively. One of the limitations of these programs is that they are currently only available on the Dartmouth College Time Sharing system using BASIC or S BASIC languages. Efforts are being made, however, to make the programs available on other computer systems for use by the Corps of Engineers Districts and Divisions. In the meantime, the computer program and the information stored can be adapted by the user to other systems. Another means of obtaining information on similar systems is to write to CRREL (TIAC) for assistance.

The purpose of this report is to present a brief description of the SEARCH and UPDATE programs and examples from each. Also, examples on how to use the stored information in the planning and design of new land treatment systems are presented in CRREL Internal Report 561.

PROGRAM SEARCH

SEARCH is the computer program being used to retrieve basic data on wastewater systems and associated papers from the computer storage bank. Figure 1 is a flow chart of the computer program SEARCH.

Each system has the same information (search parameters) encoded with it. In order to find a system, any one or more of these search parameters can be used; the computer will then look through its files and pick out all systems which fit the specifications. The searching parameters are listed below.

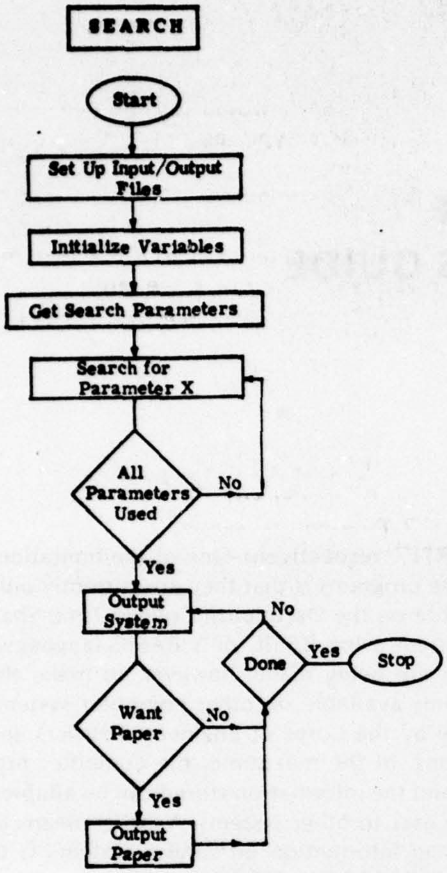


Figure 1. Flow chart of the computer program SEARCH.

Table I. System name abbreviations.

Air Force Academy	USAF ACAD.
Air Force Base	AFB
Croydon-Beddington	Croydon-BEDD.
East	E.
General	GEN.
General-irrigation	GEN.-IRR.
General-overland flow	GEN.-OVE.
General-rapid infiltration	GEN.-RAP.
General-slow infiltration	GEN.-SLO.
North	N.
Paper	PAP.
Pietmaritzburg	PIETERMARITZ.
River	R.
Saint	ST.
South	S.
University	U.
Valley	VAL.
West	W.

Searching Parameters

1. System name—the name to be used for the system. An important constraint in the program is that the system name not exceed 14 characters in length. A character is a letter, space or punctuation mark. See Table I for a list of system abbreviations.

2. Location—the location of the system, either the country (if foreign) or the state (if in the U.S.). As with the system name this must be 14 characters or less. See Table II for a list of abbreviations.

3. System type—whether the system is *operational* or a *prototype*.

4. Flow rate—treatment capacity in millions of gallons per day (mgd) of system (N/A if not available). If only a range of flow rates is given in a paper, the maximum rate is used and the flow rate range is placed in the abstract following the information about the paper.

5. Waste type—the kind of sewage used. The choices are *municipal, industrial, agricultural*, or *general* (a combination).

6. Ground cover—the vegetation, in general, present at the treatment or disposal. The choices are *grass, trees, vegetable-crop, forage-crop* and *none*. None is used if there is no ground cover present, and N/A is used if ground cover information is available.

7. Treatment type—the choices are *slow infiltration, rapid infiltration, overland flow, all*, or *N/A*.

8. Application mode—the choices are *spray, flood, both*, or *N/A*.

9. Application rate—the number of inches per week of wastewater application (to convert to cm/wk multiply by 2.54). For varied application rates, the maximum is used and the application rate range placed in the abstract following the information about the paper.

10. Purpose of the system—the choices are *treatment, irrigation, recharge*, or *general* (if multipurpose).

11. Year of initiation—the calendar year in which operations began, if available; otherwise N/A is used.

Notice that in items 4, 9 and 11 a series of numbers, a range of values or both may be entered. For instance, the following is an acceptable input for parameter 11 (year of initiation): 1967, 1932, 1951-1956, 1926. The numbers don't have to be in any particular order except when a range of values is entered (as in 1951-1956 above), in which case the first number must be

Table II. Location abbreviations.

Two-letter state abbreviations	
Alaska	AK
Alabama	AL
Arizona	AZ
Arkansas	AR
California	CA
Colorado	CO
Connecticut	CT
Delaware	DE
District of Columbia	DC
Florida	FL
Georgia	GA
Guam	GU
Hawaii	HI
Idaho	ID
Illinois	IL
Indiana	IN
Iowa	IA
Kansas	KS
Kentucky	KY
Louisiana	LA
Maine	ME
Maryland	MD
Massachusetts	MA
Michigan	MI
Minnesota	MN
Mississippi	MS
Missouri	MO
Montana	MT
Nebraska	NE
Nevada	NV
New Hampshire	NH
New Jersey	NJ
New Mexico	NM
New York	NY
North Carolina	NC
North Dakota	ND
Ohio	OH
Oklahoma	OK
Oregon	OR
Pennsylvania	PA
Puerto Rico	PR
Rhode Island	RI
South Carolina	SC
South Dakota	SD
Terinessee	TN
Texas	TX
Utah	UT
Vermont	VT
Virginia	VA
Virgin Islands	VI
Washington	WA
West Virginia	WV
Wisconsin	WI
Wyoming	WY
British Columbia	BC
Union of Soviet Socialist Republic	USSR
United States	USA

less than the second. In running the program the user first selects the desired search parameters by typing in their numbers. For example, a response of 3, 8, 11 would tell the machine to search by system type, application mode and year of initiation.

After this the computer will then ask for the information for each parameter. Note that the responses in parameters 3, 5, 6, 7, 8 and 10 may be abbreviated to the first three letters.

Example 1

Problem: Design a land treatment system for a city of 20,000 (1980) in Maryland. Most of the wastewater is municipal.

1. Assume that we would like to design a system for a town or city in Maryland. The projected population is 20,000 (in 1980). If we assume that each person uses 100 gal./day, the flow rate for this town would be 2.0 mgd.

2. The searching parameters in this case would be the location (Maryland) which is parameter 2 and the flow rate (1-3 mgd), which is parameter 4. The computer printout, using SEARCH and the indicated parameters 2 and 4, is presented in Table III.

3. If the information from the computer printout is not sufficient for the design purposes, the associated literature should be obtained and examined.

Example 2

Problem: What are the design parameters of the systems in Australia using the overland flow method of land treatment?

The searching parameters would be the Australia location (2), and the treatment type (7). The printout obtained from this search is presented in Table IV.

PROGRAM UPDATE

UPDATE is the program used to place new information into the system data bank. The program has two sections: ADD, which is used to put new systems in or add papers to systems already stored, and REVISE, which is used to change existing information. Figure 2 is a flow chart of this program.

ADD

Some preparation is required before going to the computer to add a new system. For each system the following information will be requested and should be known in advance.

Table III. A computer printout including existing systems in Maryland, wastewater flow rate less than 3 mgd.

```
SYSTEM NAME   LOCATION     SYSTEM     FLOW  WASTE TYPE   GROUND   TREATMENT        APPLIC.  APPLIC.  PURPOSE     DATE
                           TYPE       RATE               COVER    TYPE             MODE     RATE
                                      m.g.d.                                                in./wk.
------------------------------------------------------------------------------------------------------------------------

BERLIN        MD           OPERATING  0.9   INDUSTRIAL   N/A      N/A              SPRAY    N/A      TREATMENT   1953

WESTMINSTER   MD           OPERATING  2     INDUSTRIAL   N/A      N/A              N/A      N/A      TREATMENT   1972
```

Table IV. A computer printout for systems in Australia utilizing the overland flow method.

```
SYSTEM NAME   LOCATION     SYSTEM     FLOW  WASTE TYPE   GROUND   TREATMENT        APPLIC.  APPLIC.  PURPOSE     DATE
                           TYPE       RATE               COVER    TYPE             MODE     RATE
                                      m.g.d.                                                in./wk.
------------------------------------------------------------------------------------------------------------------------

MELBOURNE     AUSTRALIA    OPERATING  150   MUNICIPAL    GRASS    ALL              BOTH     N/A      TREATMENT   1897
AUTHOR-SEABROOK,B.L.
TITLE-LAND APPLICATION OF WASTEWATER IN AUSTRALIA, THE WERRIBEE FARM SYSTEM, MELBOURNE, VICTORIA
PUBLICATION-EPA, 27PP.
PUB. DATE-MAY 1975
ARE USING CROP IRRIGATION, OVERLAND FLOW AND A LAGOON SYSTEM FOR TREATMENT.

AUTHOR-SEARLE,S.S AND C.F.KIRBY
TITLE-WASTE INTO WEALTH
PUBLICATION-WATER SPECTRUM, P. 15-21
PUB. DATE-1972

AUTHOR-KEENEY,D.
TITLE-THE WERRIBEE (MELBOURNE, AUSTRALIA) SEWAGE TREATMENT FARM:  REPORT OF A VISIT BY DENNIS KEENEY ON DEC. 9,1976
PUBLICATION-
PUB. DATE-

AUTHOR-JOHNSON,R.D., R.L.JONES, T.D.HINESLY, AND D.J.DAVID
TITLE-SELECTED CHEMICAL CHARACTERISTICS OF SOILS, FORAGES, AND DRAINAGE WATER FROM THE SEWAGE FARM SERVING MELBOURNE, AUSTRALIA
PUBLICATION-DEPT. OF ARMY, CORPS OF ENGINEERS, 54PP.
PUB. DATE-JAN. 1974
```

1. System name—see Table I for a list of abbreviations
2. System location—see Table II for a list of abbreviations
3. System type
4. Flow Rate
5. Waste type
6. Ground cover
7. Treatment type
8. Application mode
9. Application rate
10. System purpose
11. Year of initiation

It should be noted that all of the above 11 questions must be answered and that the only acceptable answers to questions 3, 5, 6, 7, 8 and 10 are those that are given in the *Searching Parameters*. For questions 4, 9 and 11 N/A may be entered if the information is unknown. In addition, if reports are available, the following information should be supplied for each one:

1. Author(s) — using the following format: Last name of the principal author first, followed by his or her initials, then followed by the other authors' initials and last names (for example: Iskandar, I.K., R.S. Sletten, D.C. Leggett and T.F. Jenkins).
2. Title of the paper.
3. Publication in which report appeared, including volume number and pages if applicable.
4. Publication date: Month (if known) and year.
5. Brief abstract describing the paper (if desired).

To add a paper to a system which is already stored, the name and location of that system must be given. The computer will then print out the system data and ask if the correct system is described. If the wrong data appear, "No" should be entered and the computer will print out different system data which have the same system name and location. Once the correct

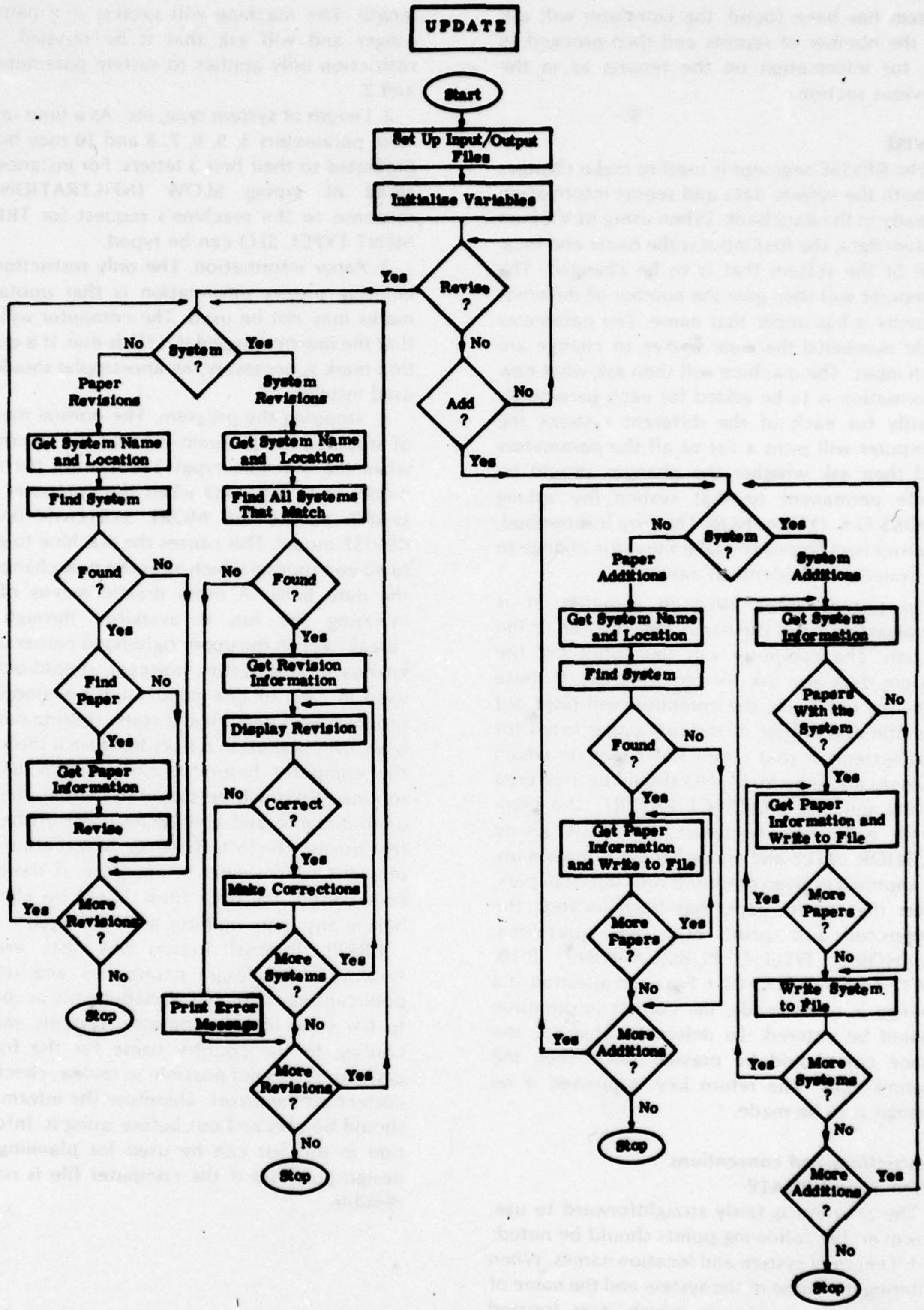


Figure 2. Flow chart of the computer program UPDATE.

system has been found, the computer will ask for the number of reports and then proceed to ask for information on the reports as in the previous section.

REVISE

The REVISE segment is used to make changes to both the system data and report information already in the data bank. When using REVISE on system data, the first input is the name and location of the system that is to be changed. The computer will then give the number of different systems it has under that name. The parameter code number(s) that one wishes to change are then input. The machine will then ask what new information is to be added for each parameter. Finally for each of the different systems the computer will print a list of all the parameters and then ask whether the changes should be made permanent to that system (by asking LOOKS O.K. (YES or NO)). Thus, by this method, the user isn't forced to make the same change to all systems with identical names.

To change information in a paper, it is necessary to input the name and location of the system. The computer will then print out the system data and ask for confirmation. If these data are not given, the computer will print out the title and author of the first paper listed for that system. If that is not the paper on which changes are to be made, NO should be answered to the question CORRECT PAPER?. The computer will then continue to repeat giving AUTHOR, TITLE and asking for confirmation until approval is given or until it runs out of papers. After the correct paper has been located, the computer will print out five questions: AUTHOR:?, TITLE:?, PUBLICATION:?, PUB. DATE:?, and ABSTRACT:?. For each question, if a change is to be made, the correct information should be entered. To delete an abstract, the space bar should be pressed once, then the "return" key. The return key is pressed if no change is to be made.

Restrictions and conventions when using UPDATE

The program is fairly straightforward to use; however, the following points should be noted:

1. Length of system and location names. When entering the name of the system and the name of the state or country in which it is located (parameters 1, and 2), the user should remember that each may not exceed 14 characters in length. The machine will protest if a name is longer and will ask that it be retyped. This restriction only applies to system parameters 1 and 2.

2. Length of system type, etc. As a time saving step, parameters 3, 5, 6, 7, 8 and 10 may be abbreviated to their first 3 letters. For instance, instead of typing SLOW INFILTRATION in response to the machine's request for TREATMENT TYPE?, SLO can be typed.

3. Paper information. The only restriction on entering proper information is that quotation marks may not be used. The computer will ask that the line be retyped if it finds one. If a quotation mark is necessary, an apostrophe should be used instead.

4. Stopping the program. The normal method of stopping the program is to hit the return key when the machine types SYSTEM (in the ADD mode) or to type NO when the computer asks WANT TO REVISE MORE SYSTEMS? (in the REVISE mode). This causes the machine to go into an end routine which adds the new changes to the data bank. A more drastic means of terminating the run is available through the "break" key in the upper right-hand corner of the keyboard. This button, however, should only be used in case of dire need—if for example the program goes berserk and starts spitting out garbage. This operation is risky because it interrupts the computer before it can go into its end routine. What will be stored in the data bank is unpredictable and everything done in the current run may be lost. However, information from previous runs is safe. In any case, if the break key is used, the data files should be checked before any more updates are attempted.

CRREL Internal Report 561 lists existing systems, their design parameters and related publications organized alphabetically according to the state for the domestic systems and according to the country name for the foreign systems. It was not possible to review, check and correct all the errors. Therefore the information should be checked out before using it. Information in this list can be used for planning and design purposes if the computer file is not accessible.

LITERATURE CITED

Iskandar, I.K. (1978) Overview of existing land treatment systems. In *State of Knowledge in Land Treatment, Proceedings, International Symposium*, Hanover, New Hampshire, August, 1978, vol. 1, p. 193-200.

Markland, R.E., L.D. Smith, J.D. Becker (1974) A benefit-cost analysis of alternative land disposal waste water methods in an urban environment. University of Missouri, St. Louis.

Pound, C.E. and R.W. Crites (1975) Treatment of municipal wastewater by land application. *Wastewater Disposal*, p. 45-56.

Wierzbicki, J. (1977) Disadvantages and advantages of sewage disposal in connection with agricultural utilization. CRREL Draft Translation 645. AD A044767.

APPENDIX A: COMPUTER LISTING OF PROGRAM SEARCH

```
SEARCH1            11/09/78            12:19:35            page 2


1010 MARGIN(133)
1020 DIM S(10), C$(10), Y$(20), Y(40,2), C(10), S$(10,7),L$(11)
1030 DIM B$(500),F(500),L(500),R(500),X(500)
1040 DIM D$(500)
1050 DIM U$(10,10)
1060 MAT Y=ZER
1070 DIM  V$(10),W$(10),X$(10),Z(10)
1080 MAT C=ZER
1090 MAT L=ZER
1100 MAT R=ZER
1110 MAT C$=NUL$
1120 MAT Y$=NUL$
1130 MAT S$=NUL$
1140 LET K=0                                  'indicator variable to show whether a search has failed
1150 LET E1=2                                 'no. of elements per record in file #1 (.---.NAM)
1160 LET E2=10                                'no. of elements per record in file #2 (.---.NUM)
1170
1180 REM CHECK TO SEE WHAT FILES WE SHOULD USE.
1190 PRINT "FOREIGN OR DOMESTIC";
1200 DO
1210    INPUT Z$
1220    LET Z$ = SEG$( Z$, 1, 3 )
1230    SELECT CASE Z$
1240    CASE "DOM"
1250         FILE #1: ".DOM.NAM"              'master file containing string information
1260         FILE #2: ".DOM.NUM"
1270         FILE #3: ".DOM.PAP"              'master file containing all paper information (term. format)
1280    CASE "FOR"
1290         FILE #1:  ".FOR.NAM"
1300         FILE #2:  ".FOR.NUM"
1310         FILE #3:  ".FOR.PAP"
1320    DEFAULT
1330         PRINT "INCORRECT FORMAT.  REENTER--";
1340    CONTINUE
1350 LOOP UNTIL Z$ = "DOM"  OR  Z$ = "FOR"
1360
1370 FILE #4:  "*"
1380 FILE #5:  "*"                            'working file to store locations of found systems
1390 'EJECT
```

```
SEARCH1            11/09/78            12:19:35            page 3


1400
1410 REM   PROCEDURE TO READ TAB SETTINGS FOR OUTPUT FORMAT
1420 FOR C=1 TO 10
1430     READ T(C)
1440 NEXT C
1450       DATA 15,29,39,45,59
1460       DATA 69,85,94,102,115
1470
1480 REM   ESTABLISH SEARCH PARAMETERS
1490 LET L$(1)="NAME OF SYSTEM"
1500 LET L$(2)="LOCATION (NAME OF STATE OR COUNTRY)"
1510 LET L$(3)="TYPE OF SYSTEM (PROTOTYPE, OPERATING)"
1520 LET L$(4)="FLOW RATE (MILLIONS OF GALLONS PER DAY, ZERO IF N/A)"
1530 LET L$(5)="TYPE OF WASTE (MUNICIPAL, INDUSTRIAL, AGRICULTURAL, GENERAL)"
1540 LET L$(6)="CHARACTERISTIC GROUND COVER (GRASS, TREES, VEG.-CROP, FORAGE, NONE, N/A)"
1550 LET L$(7)="TREATMENT TYPE (SLOW INFILT., RAPID INFILT., OVERLAND FLOW, ALL, N/A)"
1560 LET L$(8)="APPLICATION MODE (SPRAY, FLOOD, BOTH, N/A)"
1570 LET L$(9)="APPLICATION RATE (INCHES PER WEEK ZERO IF N/A)"
1580 LET L$(10)="PURPOSE OF SYSTEM (TREATMENT, IRRIGATION, RECHARYGE, GENERAL)"
1590 LET L$(11)="YEAR OF INITIATION"
1600 'EJECT
```

```
SEARCH1           11/09/78          12:19:43          page 4


1610 REM   ESTABLISH ANSWERS TO PARAMETERS #3,5,6,7,8,10
1620 READ S$(3,1), S$(3,2), S$(3,3)
1630      DATA "2", "PROTOTYPE", "OPERATING"
1640 READ S$(5,1), S$(5,2), S$(5,3), S$(5,4), S$(5,5)
1650      DATA "4", "MUNICIPAL", "INDUSTRIAL", "AGRICULTURAL", "GENERAL"
1660 READ S$(6,1), S$(6,2), S$(6,3), S$(6,4), S$(6,5), S$(6,6), S$(6,7)
1670      DATA "6", "GRASS", "TREES", "VEG.-CROP", "FORAGE", "NONE", "N/A"
1680 READ S$(7,1), S$(7,2), S$(7,3), S$(7,4), S$(7,5), S$(7,6)
1690      DATA "5", "SLOW INFILT.", "RAPID INFILT.", "OVERLAND FLOW", "ALL", "N/A"
1700 READ S$(8,1), S$(8,2), S$(8,3), S$(8,4), S$(8,5)
1710      DATA "4", "SPRAY", "FLOOD", "BOTH", "N/A"
1720 READ S$(10,1), S$(10,2), S$(10,3), S$(10,4), S$(10,5)
1730      DATA "4", "TREATMENT", "IRRIGATION", "RECHARGE", "GENERAL"
1740
1750
1760 IF ( INT( (LOF(#2) - 1)/10 ) <> (LOF(#2) - 1)/10 )
1770 THEN PRINT "START-UP ERROR: FILE #2 (.---.NUM) NOT OF CORRECT LENGTH."
1780      STOP
1790 CONTINUE
1800
1810
1820 REM    INPUT & CHECKOUT OF SEARCH PARAMETER NUMBERS
1830 PRINT "SEARCH PARAMETERS (TYPE 0 FOR A LIST)";
1840 DO
1850    LET C9=0                                 'flag variable for incorrect input
1860    MAT S=ZER
1870    MAT INPUT S                              'search parameters
1880    LET N=NUM                                'number of search parameters
1890    FOR X=1 TO N
1900        IF S(X)<>INT(S(X)) OR S(X)>11 OR S(X)<1     'check for correct format
1910        THEN IF S(X)=0                       'if a list is desired
1920             THEN PRINT
1930                  PRINT "THE POSSIBLE PARAMETERS ARE:"
1940                  FOR X1=1 TO 11             'print the list of parameters
1950                      PRINT "     ";X1;"-";L$(X1)
1960                  NEXT X1
1970                  PRINT "ENTER SEARCH PARAMETERS";
1980                  LET X=N
1990                  LET C9=1
2000             ELSE PRINT "NO SUCH PARAMETER AS:", S(X)
2010                  PRINT "TRY RETYPING IT";
2020                  INPUT S(X)
2030                  LET X=X-1
2040             CONTINUE
2050        CONTINUE
2060    NEXT X
2070 LOOP WHILE C9=1
2080 'EJECT
```

```
SEARCH1           11/09/78          12:19:51          page 5


2090 REM    SEARCH  PARAMETER INPUT & SEARCH ROUTINES
2100 LET X=1                                     'counter
2110 DO
2120    PRINT L$(S(X));                          'print heading
2130    IF END #4                                'is file #4 empty?
2140    THEN LET F1=1                            '      yes (no systems have been found yet)
2150    ELSE LET F1=0                            '      no (file contains locations of found systems)
2160    CONTINUE
2170
2180    SELECT CASE S(X)
2190
2200
2210        REM    SYSTEM NAME & LOCATION
2220    CASE 1,2
2230        INPUT C$(S(X))                       'input system name or location
2240
2250        RESET #1:0                           'reset .---.NAM
2260        LET F=1
2270        PERFORM ENDCHECK
2280        DO WHILE S=1
2290           IF F1<>1                          'was file #4 empty at the start of this routine?
2300           THEN INPUT #4:P                   'if not, pick a system location from file #4
2310                RESET #1:(P-1)*E1            'reset .---.NAM to pick up that system
2320           CONTINUE
2330           READ #1:D$(1), D$(2)              'readout the system name & location from .---.NAM
2340           IF C$(S(X))=D$(S(X))              'if the existing system matches the desired system
2350           THEN PRINT #5:LOC(#1)/E1          'store its location
2360           CONTINUE
2370           LET F=1                           'file no. to be examined in ENDCHECK (1=.---.NAM)
2380           PERFORM ENDCHECK                  'check whether we're done (if so, set S equal to 0)
2390        LOOP
2400        PERFORM RESET                        'dump results from file #5 to #4 & reset #4
2410        'EJECT
```

```
SEARCH1          11/09/78          12:19:59          page 6


2420        REM     SYSTEM TYPE, WASTE TYPE, GROUND COVER, ETC.
2430     CASE 3,5,6,7,8,10
2440        LET Z=S(X)-1                   'set up numbering system compatable with .---.NUM
2450        DO
2460           INPUT A$                    'input search parameter
2470           LET A1$=SEG$(A$,0,3)        'abbreviate it to the first 3 letters
2480           LET X1=VAL(S$(S(X),1))+1    'location of last acceptable answer
2490           LET X2=2                    'location of first posible answer
2500           DO
2510              IF A1$=SEG$(S$(S(X),X2),0,3) THEN LET C(Z)=X2-1    'if answer is acceptable, store its number code
2520              LET X2=X2+1
2530           LOOP UNTIL X2>X1 OR C(Z)>0
2540           IF C(Z)=0                   'if no acceptable answer was found
2550           THEN PRINT  "NO SUCH SPECIFICATION AS:"; A$
2560                PRINT "THE POSSIBILITIES ARE:"
2570                FOR X2=2 TO X1
2580                    PRINT "      "; S$(S(X),X2)     'give the list of acceptable answers
2590                NEXT X2
2600                PRINT "TRY RETYPING IT";
2610           CONTINUE
2620        LOOP UNTIL C(Z)>0
2630
2640
2650        RESET #2:0                     'reset .---.NUM
2660        LET F=2                        'file no. to be examined in ENDCHECK (2=.---.NUM)
2670        DO
2680           IF F1<>1                    'was file #4 empty at the start of this routine?
2690           THEN INPUT #4:P             'if not, pick a system location from file #4
2700                RESET #2:(P-1)*E2      'reset .---.NUM to pick up that system
2710           CONTINUE
2720           READ #2:D(1), D(2), D(3), D(4), D(5), D(6), D(7), D(8), D(9), D(10)  'read out all system parameters
2730           IF C(Z)=D(Z) THEN PRINT #5: LOC(#2)/E2      'if a match on the parameter you're searching under, store its location
2740           PERFORM ENDCHECK            'check to see if done (if so let S=0)
2750        LOOP WHILE S=1
2760        PERFORM RESET                  'dump results from file #5 into #4 &reset #4
2770        'EJECT
```

```
SEARCH1          11/09/78          12:20:09          page 7


2780        REM     FLOW RATE, APPLIC. RATE, DATE
2790     CASE 4,9,11
2800        LET Z=S(X)-1                         'set up numbering system compatable with .---.NUM
2810        LET F=2                              'file no. to be examined in ENDCHECK (2=.---.NUM)
2820        MAT INPUT Y$                         'enter all the years, rates, etc. that will be used for searching
2830        LET Y1=NUM                           'no. of distinct entries (example: 1920, 1938-1952 would be 2)
2840        FOR X1=1 TO Y1                       'loop to separate entries into distinct numbers
2850           LET P=POS(Y$(X1),"-",0)           'position of dash if present (as in 1938-1952)
2860           IF P>0                            'if there is a dash
2870           THEN LET C$=SEG$(Y$(X1),0,P-1)    'first number (string value)
2880                LET D$=SEG$(Y$(X1),P+1,LEN(Y$(X1)))    'second number (string value)
2890                LET C=VAL(C$)                'first number (numeric value)
2900                LET D=VAL(D$)                'second number (numeric value)
2910                LET Y(X1,1)=C                'first number
2920                LET Y(X1,2)=D-C              'difference btween first & second numbers
2930           ELSE LET Y(X1,1)=VAL(Y$(X1))      'there is only one value, convert to numeric
2940           CONTINUE
2950        NEXT X1
2960
2970        RESET #2:0                           'reset .---.NUM
2980        DO
2990           IF F1<>1                          'was file #4 empty at the start of this routine?
3000           THEN INPUT #4:P                   'if not, pick a system location from file #4
3010                RESET #2:(P-1)*E2            'reset .---.NUM to pick up that system
3020           CONTINUE
3030           READ #2:D(1), D(2), D(3), D(4), D(5), D(6), D(7), D(8), D(9), D(10)  'read out all system parameters
3040           LET X1=1                          'counter
3050           LET C7=0                          'flag variable to indicate a find (0 indicates no find)
3060           DO WHILE C7=0 AND X1 <= Y1
3070              IF D(Z)<=Y(X1,1)+Y(X1,2)       'test whether value from .---.NUM is less than the upper bound
3080              THEN IF D(Z)>=Y(X1,1)          'test whether value from .---.NUM is greater than lower bound
3090                   THEN PRINT #5:LOC(#2)/E2  'store its location
3100                        LET C7=1             'reset flag variable
3110                   ELSE LET X1 = X1 + 1
3120                   CONTINUE
3130              ELSE LET X1=X1+1               'otherwise go to the next value inputed
3140              CONTINUE
3150           LOOP
3160           PERFORM ENDCHECK                  'check to see if we're done & set S equal to 0 if so
3170        LOOP WHILE S=1
3180        PERFORM RESET                        'dump results from file #5 into file #4 & reset #4
3190
3200
3210     CONTINUE
3220     LET X=X+1                               'go to the next parameter
3230 LOOP UNTIL X>N OR K=1                       'K=1 indicates a search  failed & thus terminates program
3240 'EJECT
```

```
SEARCH1          11/09/78          12:20:18          page 8

3250 REM      THIS SECTION ORDERS THE RANDOM ACCESSS ADDRESSESS OF THE FOUND SYSTEMS( TO THIS POINT
3260 REM      UNORDERED AND IN FILE #4) .  THE ADDRESSESS ARE REORDERED ALPHABETICALLY ACCORDING
3270 REM      TO THE STATE (OR COUNTRY) OF THEIR RESPECTIOVE SYSTEMS.  THE ORDERING PROCESS IS
3280 REM      ACHEIVED BY A STANDARD BINARY TREE SORT.
3290 LET N=0
3300 DO WHILE MORE #4
3310     INPUT #4: X
3320     RESET #1: (X-1)*E1
3330     READ #1: C1$,C$
3340     LET N=N+1
3350     CALL "BILDTREE": B$(),L(),R(),C$,N,X,X()  'build an alphabetically ordered tree(left son<father<right son)
3360 LOOP
3370
3380
3390 CALL "TREESORT": L(),R(),1,1,F(),X()     'send completed tree x() to tree sort, return the ordered vector f()
3400 'EJECT
```

```
SEARCH1          11/09/78          12:20:26          page 9

3410 REM      F IS A VECTOR OF RANDOM ACCESS ADDRESSESS OF FOUND SYSTEMS, ALPHABETICALLY ORDERED
3420 REM      BY STATE.  IN THIS SECTION, THE ADDRESSESS OF THE SYSTEMS WITHIN EACH STATEARE
3430 REM      ORDERED ALPHABETICALLY.  A STANDARD BINARY TREESORTIS AGAIN USED FOR THE
3440 REM      ORDERING PROCESS.T
3450 MAT L=ZER
3460 MAT R=ZER
3470 MAT D$=NUL$
3480 LET L(0)=0
3490 LET R(0)=0
3500
3510 LET I=0                                'count of systems read from vector F
3520 LET K=0                                'count of systems ordered by system name
3530 PERFORM GET NAMEDATA                   'get a system
3540 DO UNTIL B$(I)="X"                     'end of file marker
3550     DO
3560        CALL "BILDTREE": D$(),L(),R(),D$(I),I,X,X()      'build an alphabetically ordered tree(left son<father<right son)
3570        PERFORM GET NAMEDATA
3580     LOOP WHILE B$(I)=B$(I-1)            'i.e. loop until all systems of the same state have been read
3590     IF B$(I-1)<>B$(I-2)                 'if there is only one system for the state
3600     THEN LET F(I-1)=X(I-1)              'then do not bother sorting tree.there is only one system from that state
3610     ELSE LET L=K+1                      'else un,load the tree composed of all systems in one state
3620        LET M=K+1
3630        CALL "TREESORT": L(),R(),M,L,F(),X()
3640     CONTINUE
3650
3660     LET K=I-1                           'save the vector F address of the last sys in a particular satae
3670     MAT L=ZER
3680     MAT R=ZER
3690     LET L(0)=0
3700     LET R(0)=0
3710     LET D$(1)=" "
3720 LOOP
3730
3740
3750 REM      THE VECTOR OF ALPHABETICALLY ORDERED ADDRESSES ARE READ BACK INTO
3760 REM      FILE #4
3770 SCRATCH #4
3780 FOR K=1 TO N
3790     PRINT #4: F(K)
3800 NEXT K
3810 RESET #4
3820 'EJECT
```

```
SEARCH1             11/09/78            12:20:35            page 10


3830 REM    READOUT OF RESULTS
3840 LET V=0                                            'count of lines  (already printed ))
3850 LET V9=0                                           'count of lines to be printed
3860 IF K<>0
3870 THEN PRINT
3880       PRINT "DO YOU WISH A READOUT OF AVAILABLE PAPERS";
3890       INPUT R$
3900       PRINT CHR$(ASC(FF))                          'form feed
3910
3920
3930       REM     PRINT HEADERS
3940       PRINT
3950       PRINT "SYSTEM NAME"; TAB(T(1)); "LOCATION"; TAB(T(2)); "SYSTEM";
3960       PRINT TAB(T(3)); "FLOW"; TAB(T(4)); "WASTE TYPE"; TAB(T(5)); "GROUND";
3970       PRINT TAB(T(6)); "TREATMENT"; TAB(T(7)); "APPLIC."; TAB(T(8)); "APPLIC.";
3980       PRINT TAB(T(9)); "PURPOSE"; TAB(T(10)); "DATE"
3990       PRINT TAB(T(2)); "TYPE"; TAB(T(3)); "RATE"; TAB(T(5)); "COVER";
4000       PRINT TAB(T(6)); "TYPE"; TAB(T(7)); "MODE"; TAB(T(8)); "RATE"
4010       PRINT TAB(T(3)); "m.g.d."; TAB(T(8)); "in./wk."
4020       PRINT
4030       PRINT
4040       LET V9=7
4050
4060
4070       REM     READOUT  OF FOUND SYSTEMS
4080
4090       DO WHILE MORE #4
4100          INPUT #4:X
4110          RESET #1:(X-1)*E1
4120          RESET #2:(X-1)*E2
4130          LET C3$=C2$
4140          READ #1:C1$, C2$
4150          READ #2:D(1), D(2),D(3), D(4), D(5), D(6), D(7), D(8), D(9), D(10)
4160          IF C3$<> C2$                     'if state (or country) of adjoining system is different
4170          THEN PERFORM LINE
4180                LET V9=V9+1
4190          CONTINUE
4200          PRINT
4210          PRINT
4220          LET V9=V9+7
4230
4240
4250          REM     READOUT  OF PAPER INFORMATION
4260          IF R$="YES"
4270          THEN IF D(1)>0                   'D(1) is the paper location pointer in .---.NUM, 0 indicates no papers
4280                THEN RESET #3
4290                      LET V1=0
4300                      DO
4310                         INPUT #3:A,          'A is the paper location pointer in .---.PAP
4320                         IF A=D(1)
4330                         THEN LET V1=V1+1    'count of papers
4340                               INPUT #3:V$(V1),W$(V1),X$(V1),E$(V1),Z(V1),        'author,title,publication,date,no. lines of abstract
4350                               FOR V3= 1 TO Z(V1)
4360                                     INPUT #3: U$(V1,V3),
```

```
SEARCH1             11/09/78            12:20:44            page 11

4370                               NEXT V3
4380                               LET V9=V9+Z(V1)+5
4390                         ELSE LINPUT #3: Z$        'in effect, a carriage return
4400                         CONTINUE
4410                      LOOP WHILE MORE #3
4420                CONTINUE
4430          CONTINUE
4440          'EJECT
```

```
SEARCH1           11/09/78          12:20:44          page 12

4450
4460
4470         REM     PRINTOUT OF FOUND SYSTEMS
4480         IF V9+V>55
4490         THEN LET V=V9                       'reset line counter
4500              LET V9=0
4510              PRINT CHR$(ASC(FF))            'form feed
4520         ELSE PRINT                          'print a line
4530              LET V=V+V9
4540              LET V9=0
4550         CONTINUE
4560         PRINT C1$; TAB(T(1)); C2$;          'dump out the system
4570         FOR X1=2 TO 10
4580             PRINT TAB(T(X1));
4590             SELECT CASE X1
4600             CASE 3, 8, 10
4610                  IF D(X1)<>0                'O signals that the category is not applicable or unknown
4620                  THEN PRINT D(X1);
4630                  ELSE PRINT " N/A ";
4640                  CONTINUE
4650             DEFAULT
4660                  PRINT S$(X1+1, 1+D(X1));
4670             CONTINUE
4680         NEXT X1
4690
4700         PRINT
4710
4720         REM     PRINTOUT OF PAPERS OF FOUND SYSTEMS
4730         IF R$="YES"
4740         THEN PRINT
4750              FOR V2 = 1 TO V1
4760                  PRINT "AUTHOR-"; W$(V2)
4770                  PRINT "TITLE-"; V$(V2)
4780                  PRINT "PUBLICATION-"; X$(V2)
4790                  PRINT "PUB. DATE-"; E$(V2)
4800
4810                  REM     READOUT & PRINTING OF ABSTRACT
4820                  FOR V3= 1 TO Z(V2)
4830                      PRINT U$(V2, V3)
4840                  NEXT V3
4850                  PRINT
4860              NEXT V2
4870         CONTINUE
4880      LOOP
4890 ELSE PRINT "COULDN'T FIND ANYTHING THAT FITS THOSE SPECIFICATIONS"
4900 CONTINUE
4910 PRINT CHR$(ASC(FF))
4920 'EJECT
```

```
SEARCH1           11/09/78          12:20:54          page 13

4930 DEFINE ENDCHECK                          'routine to test for the end of ---.(NUM OR NAM) or file #4 where applicable
4940      IF F1<>1                            'if true then we're using file #4 as a key
4950      THEN IF END #4                      'test whether #4 is through
4960           THEN LET S=0                   'yes
4970           ELSE LET S=1                   'no
4980           CONTINUE
4990      ELSE IF LOC(#F)>=LOF(#F)-(F-1)      'otherwise, test for the end of Master File (F=1:.---.NAM, F=2:.---.NUM)
5000           THEN LET S=0                   'yes, file has ended
5010           ELSE LET S=1                   'no, it hasn't
5020           CONTINUE
5030      CONTINUE
5040 DEFEND
5050
5060
5070 DEFINE RESET                             'procedure to dump file #5 into #4 & reset #4
5080      SCRATCH #4
5090      RESET #5
5100      DO WHILE MORE #5                    'loop to dump #5 into #4
5110         INPUT #5:A
5120         PRINT #4:A
5130      LOOP
5140      SCRATCH #5
5150      RESET #4
5160      IF END #4 THEN LET K=1              'search turned up nothing
5170 DEFEND
5180
5190 DEFINE LINE
5200      FOR X1=1 TO 133
5210          PRINT "-";
5220      NEXT X1
5230      PRINT
5240 DEFEND LINE
5250
5260 DEFINE GET NAMEDATA
5270      LET I=I+1                           'increase the count of processed systems by one
5280      IF I=N+1                            'if the count of processed systems is greater than the actual number
5290      THEN LET B$(I)="X"                  'then set the flag
5300      ELSE LET X=F(I)                     'else get the actual data
5310           RESET #1:  (X-1)*E1
5320           READ #1:  D$(I),B$(I)
5330      CONTINUE
5340 DEFEND
5350 'EJECT
```

```
5360 END
5370 SUB "BILDTREE": B$(),L(),R(),C$,N,X,X()       'adds a new word to ordered binary tree
5380     LET J=0                                 'vector address
5390     DO
5400        LET J1=J
5410        IF C$ < B$(J)                         'is word less than the word already at b$(j)?
5420        THEN LET J=L(J)                       'then get the address of hte left son of word at b$(j)
5430             LET D=0                          'set the flag
5440        ELSE LET J=R(J)
5450             LET D=1
5460        CONTINUE
5470     LOOP UNTIL J=0                           'continue until an empty node has been found
5480     LET B$(N)=C$                             'add the new word to the tree
5490     LET X(N)=X                               'set up the vector of addresses
5500     IF D=0
5510     THEN LET L(J1)=N                         'set up the link from the father to the new left son
5520     ELSE LET R(J1)=N                         'set up the link from the father to the new right son
5530     CONTINUE
5540 SUBEND
5550
5560
5570 REM     THIS SUBPROG USES RECURSION TO SORT THE ALPHA-ORDERED ADDRESSES IN THE TREE.
5580 SUB "TREESORT": L(),R(),I,J,F(),X()      ' return the alpha-ordered vector F
5590     IF L(I)>0 THEN CALL "TREESORT":L(),R(),L(I),J,F(),X()   'IF THERE IS A LEFT  SON, THEN CALL THE SUBPROG
5600     LET F(J)=X(I)                            'this is a terminal node (if a left son exists, it is in vector F)
5610     LET J=J+1                                'increase the count of elements in vector F
5620     IF R(I)> 0 THEN CALL "TREESORT":L(),R(),R(I),J,F(),X()  'if there is a right son, then call the subprog
5630 SUBEND

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

TERMINUS     SEARCH1

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
12:21:19                                                                                                              11/09/78
```

APPENDIX B: COMPUTER LISTING OF PROGRAM UPDATE

```
HDG18501      UPDATE1

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

12:21:29                                                                                                                    11/09/78

1000 '
```

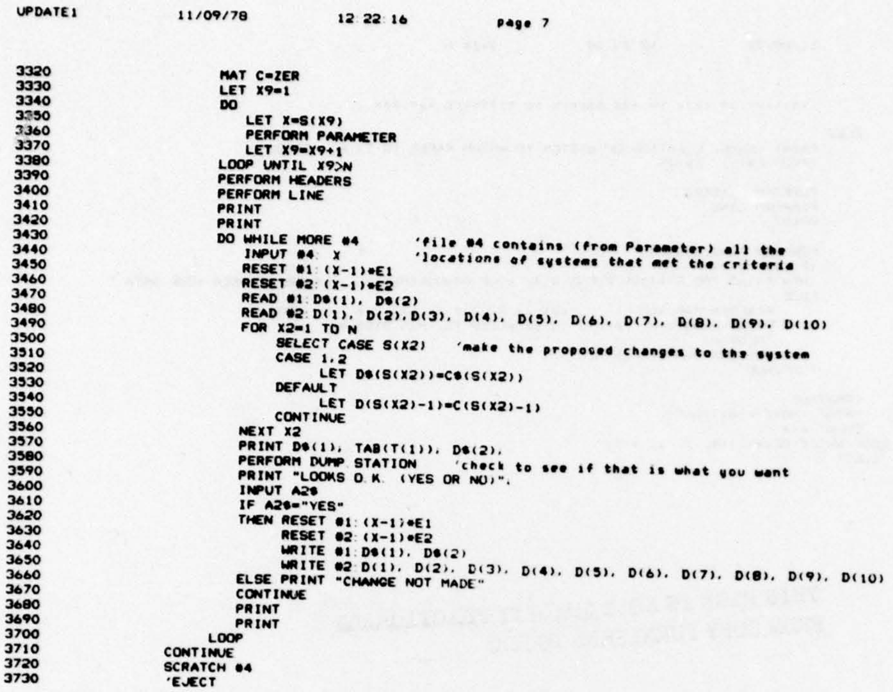
```
UPDATE1           11/09/78            12:21:34           page 2


1010 ' SBASIC PROGRAM TO UPDATE THE EXISTING SYSTEMS FILE
1020 ' CAN TYPE IN FIRST 3 LETTERS FOR ALL COMMANDS
1030 ' LONGEST POSSIBLE SYST. NAMES & LOCATIONS ARE 14 CHARACTERS.
1040
1050
1060
1070
1080 MARGIN(133)
1090 DIM C$(10), C(10), S$(10,7), R(10), L$(12), M$(12), D(10)
1100 DIM S(12)
1110 MAT C$=NUL$
1120 MAT S$=NUL$
1130 LET Z$=CHR$(ASC("))                        'an admittedly roundabout way of getting a "
1140
1150 REM CHECK TO SEE WHICH FILES WE SHOULD USE
1160 PRINT "FOREIGN OR DOMESTIC";
1170 DO
1180     INPUT Z5$
1190     LET Z5$ = SEG$( Z5$, 1, 3 )
1200     SELECT CASE Z5$
1210     CASE "DOM"
1220          FILE #1: ".DOM.NAM,DOM"            'rand. access master file which contains string info. (name & location)
1230          MARGIN#1: 15                       'longest permissable string in .---.NAM
1240          FILE #2: ".DOM.NUM,DOM"            'file where numeric info on system is kept
1250          FILE #3:  ".DOM.PAP,DOM"           'terminal format file which contains info. on papers
1260          MARGIN#3:0
1270     CASE "FOR"
1280          FILE #1:  ".FOR.NAM,FOR"
1290          MARGIN #1:  15
1300          FILE #2:  ".FOR.NUM,FOR"
1310          FILE #3:  ".FOR.PAP,FOR"
1320          MARGIN #3:  0
1330     DEFAULT
1340          PRINT "INCORRECT FORMAT.  REENTER CATAGORY--";
1350     CONTINUE
1360 LOOP UNTIL Z5$ = "DOM"  OR  Z5$ = "FOR"
1370
1380 LET L9 = LOF( #2 )
1390 IF ( INT(( L9 - 1 )/10 ) <> ( L9 - 1 )/10 ')
1400 THEN PRINT ".---.NUM IS MESSED UP AGAIN. DO NOT USE THIS PROGRAM UNTIL IT IS FIXED."
1410        STOP
1420 CONTINUE
1430
1440 FILE #4:  "*"
1450 FILE #5:  "*"                                'transfere scratch file(see below).
1460 MARGIN #5:0
1470 SCRATCH #5                                   'there goes the back-up data
1480 LET E1=2                                     'length of logical entry in .---.NAM
1490 LET E2=10                                    'length of logical entry in .---.NUM
1500 'EJECT
```

```
UPDATE1            11/09/78            12:21:42            page 3


1510 RESET #1:LOF(#1)
1520 RESET #2: LOF(#2)-1
1530 READ #2: P                                      'read last no. in ---.NUM which is the last paper location no. handed out
1540 RESET #2:LOF(#2)-1
1550
1560
1570 REM   PROCEDURE TO READ TAB SETTINGS FOR OUTPUT FORMAT
1580 FOR C=1 TO 10
1590      READ T(C)
1600 NEXT C
1610       DATA 16, 31, 43, 51, 66              'tab settings
1620       DATA 78, 94, 104, 114, 127           'tab settings
1630
1640
1650 REM   ESTABLISH SEARCH PARAMETERS
1660 LET L$(1)="NAME OF SYSTEM"
1670 LET L$(2)="LOCATION (NAME OF STATE OR COUNTRY)"
1680 LET L$(3)="TYPE OF SYSTEM (PROTOTYPE, OPERATING)"
1690 LET L$(4)="FLOW RATE (MILLIONS OF GALLONS PER DAY, ZERO IF N/A)"
1700 LET L$(5)="TYPE OF WASTE (MUNICIPAL, INDUSTRIAL, AGRICULTURAL, GENERAL)"
1710 LET L$(6)="CHARACTERISTIC GROUND COVER (GRASS, TREES, VEG.-CROP, FORAGE, NONE, N/A)"
1720 LET L$(7)="TREATMENT TYPE (SLOW INFILT., RAPID INFILT., OVERLAND FLOW, ALL, N/A)"
1730 LET L$(8)="APPLICATION MODE (SPRAY, FLOOD, BOTH, N/A)"
1740 LET L$(9)="APPLICATION RATE (INCHES PER WEEK, ZERO IF N/A)"
1750 LET L$(10)="PURPOSE OF SYSTEM (TREATMENT, IRRIGATION, RECHARYGE, GENERAL)"
1760 LET L$(11)="YEAR OF INITIATION"
1770 LET L$(12)="NO. OF PAPERS (0 IF NONE)"
1780
1790 REM   ESTABLISH ANSWERS TO PARAMETERS #3,5,6,7,8,10
1800 READ S$(3,1), S$(3,2), S$(3,3)
1810       DATA "2", "PROTOTYPE", "OPERATING"
1820 READ S$(5,1), S$(5,2), S$(5,3), S$(5,4), S$(5,5)
1830       DATA "4", "MUNICIPAL", "INDUSTRIAL", "AGRICULTURAL", "GENERAL"
1840 READ S$(6,1), S$(6,2), S$(6,3), S$(6,4), S$(6,5), S$(6,6), S$(6,7)
1850       DATA "6", "GRASS", "TREES", "VEG.-CROP", "FORAGE", "NONE", "N/A"
1860 READ S$(7,1), S$(7,2), S$(7,3), S$(7,4), S$(7,5), S$(7,6)
1870       DATA "5", "SLOW INFILT.", "RAPID INFILT.", "OVERLAND FLOW", "ALL", "N/A"
1880 READ S$(8,1), S$(8,2), S$(8,3), S$(8,4), S$(8,5)
1890       DATA "4", "SPRAY", "FLOOD", "BOTH", "N/A"
1900 READ S$(10,1), S$(10,2), S$(10,3), S$(10,4), S$(10,5)
1910       DATA "4", "TREATMENT", "IRRIGATION", "RECHARGE", "GENERAL"
1920
1930 REM   ABREVIATED FORMS TO BE USED IN QUESTIONS
1940 READ M$(1), M$(2), M$(3), M$(4), M$(5), M$(6)
1950 READ M$(7), M$(8), M$(9), M$(10), M$(11), M$(12)
1960       DATA "SYSTEM", "LOCATION", "SYSTEM TYPE", "FLOW RATE", "WASTE TYPE", "GROUND COVER"
1970       DATA "TREAT. TYPE", "APPLIC. MODE", "APPLIC. RATE", "PURPOSE", "DATE", "# OF PAPERS"
1980
1990 'EJECT
```

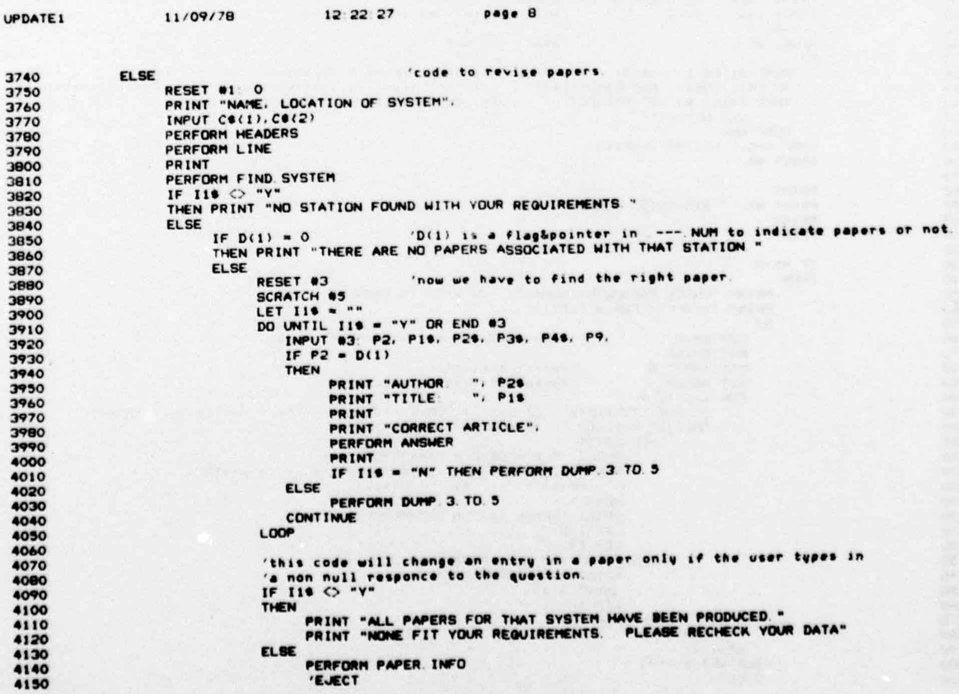
```
UPDATE1            11/09/78            12:21:51            page 4


2000 PRINT "DO YOU WISH TO 'ADD' NEW INFO. OR 'REVISE' OLD";
2010 DO
2020     INPUT I$
2030     LET I$ = SEG$(I$,1,3)
2040     IF I$ <> "ADD" AND I$ <> "REV"
2050     THEN PRINT "INCORRECT COMMAND   'ADD' OR 'REVISE'";
2060     CONTINUE
2070 LOOP UNTIL I$ = "ADD" OR I$ = "REV"
2080 IF I$="ADD"
2090 THEN
2100       DO
2110           PRINT "DO YOU WANT TO ADD A 'SYSTEM' OR A 'PAPER'";
2120           INPUT I$
2130           IF SEG$( I$, 1, 3 ) = "SYS"
2140           THEN
2150                 PRINT "DO YOU NEED A LIST OF THE CATAGORIES";
2160                 PERFORM ANSWER
2170                 IF I1$="Y"
2180                 THEN PRINT
2190                       PRINT "YOU WILL BE ASKED TO PROVIDE THE FOLLOWING INFORMATION"
2200                       FOR X1=1 TO 12
2210                           PRINT "      ";X1;"-";L$(X1)
2220                       NEXT X1
2230                       PRINT "FOR CATAGORIES #4, 9 & 11 (FLOW RATE, APPLIC. RATE"
2240                       PRINT "AND YEAR), IF THE INFORMATION IS NOT KNOWN OR NOT APPLICABLE"
2250                       PRINT "TYPE A 0 (ZERO) "
2260                       PRINT "YOU WILL THEN BE ASKED TO INPUT INFORMATION ON ANY PAPERS "
2270                       PRINT "DOING ONE PAPER AT A TIME THE MACHINE WILL REQUEST THE TITLE,"
2280                       PRINT "THE AUTHOR(S), PUBLICAIION WHERE II APPEARED, YEAR OF "
2290                       PRINT "PUBLICATION AND AN ABSTRACT "
2300                       PRINT "WHEN YOU ARE THROUGH ENTERING DATA, HIT RETURN WHEN THE "
2310                       PRINT "MACHINE TYPES 'SYSTEM?'"
2320                 CONTINUE
2330
2340                 PRINT
2350                 PRINT
2360                 DO
2370                    MAT C=ZER
2380                    LET X=1                     'counter
2390                    DO
2400                       PERFORM PARAMETER
2410                       LET X=X+1
2420                    LOOP UNTIL X>12 OR C$(1)=""   'when X2=12 then P2 is returned by 'Parameter'
2430
2440                    IF C$(1)>"" AND H<>1     'means that we have to record an addition.( Its name is
2450                    THEN RESET #2: LOF(#2) - 1   'not null and we haven't seen it before.)
2460                         WRITE #1: C$(1), C$(2)   'name, then location
2470                         WRITE #2: P2, C(2), C(3), C(4), C(5), C(6), C(7), C(8), C(9), C(10)
2480                         WRITE #2: P
490                        PRINT
2500                    CONTINUE
2510                 LOOP WHILE C$(1)>"" AND H <> 1  'terminate if either 1. H=1, or 2. C$(1) is null
2520                 'EJECT
```

```
UPDATE1             11/09/78            12:21:59          page 5


2530                'section of code to add papers to existing systems
2540                '
2550           ELSE
2560                PRINT "NAME, LOCATION OF SYSTEM TO WHICH PAPER IS TO BE ADDED",
2570                INPUT C$(1), C$(2)
2580
2590                PERFORM HEADERS
2600                PERFORM LINE
2610                PRINT
2620
2630                PERFORM FIND SYSTEM
2640                IF I1$ <> "Y"
2650                THEN PRINT "NO STATION FOUND WITH YOUR REQUIREMENTS   PLEASE RECHECK YOUR DATA "
2660                ELSE
2670                     PERFORM PAP NUM        'set up P2 and pointer in  --- NUM
2680                     PRINT "HOW MANY PAPERS TO BE ADDED TO THIS SYSTEM",
2690                     INPUT P1
2700                     PERFORM MANY PAPER ADD
2710                CONTINUE
2720
2730           CONTINUE
2740           PRINT "MORE ADDITIONS",
2750           INPUT I1$
2760      LOOP WHILE SEG$( I1$, 1, 1) = "Y"
2770      'EJECT
```

```
UPDATE1             11/09/78            12:22:08          page 6


2780 ELSE
2790      DO
2800           PRINT "DO YOU WANT TO REVISE A 'SYSTEM' OR A 'PAPER'",
2810           INPUT Z5$
2820           LET Z5$ = SEG$(Z5$,1,3)
2830           IF Z5$ = "SYS"
2840           THEN
2850                PRINT "NAME, LOCATION OF SYSTEM TO BE CHANGED",
2860                INPUT C$(1), C$(2)           'input system name and location
2870                LET N1=0
2880                RESET #1:0                   'reset  --- NAM
2890                DO
2900                   READ #1 D$(1), D$(2)      'readout the system name & location from  --- NAM
2910                   IF C$(1)=D$(1) AND C$(2)=D$(2)    'if the existing system matches the desired system
2920                   THEN PRINT #4 LOC(#1)/E1       'store its location
2930                        LET N1=N1+1
2940                   CONTINUE
2950                LOOP WHILE LOC(#1)<LOF(#1)
2960                RESET #4
2970
2980                PRINT
2990                PRINT N1, " SYSTEM(S) FOUND"
3000                PRINT
3010                PRINT
3020
3030                IF N1>0
3040                THEN
3050                     PRINT "ENTER PARAMETER NUMBERS YOU WISH TO CHANGE"
3060                     PRINT "(TYPE 0 FOR A LIST)",
3070                     DO
3080                        LET C9=0           'flag variable for incorrect input
3090                        MAT S=ZER
3100                        MAT INPUT S        'search parameters
3110                        LET N=NUM          'number of search parameters
3120                        FOR X=1 TO N
3130                             IF S(X)<>INT(S(X)) OR S(X)>11 OR S(X)<1      'check for correct format
3140                             THEN IF S(X)=0       'if a list is desired
3150                                  THEN PRINT
3160                                       PRINT "THE POSSIBLE PARAMETERS ARE "
3170                                       FOR X1=1 TO 11     'print the list of parameters
3180                                           PRINT "     ",X1;"-";L$(X1)
3190                                       NEXT X1
3200                                       PRINT "ENTER SEARCH PARAMETERS",
3210                                       LET X=N
3220                                       LET C9=1
3230                                  ELSE PRINT "NO SUCH PARAMETER AS ", S(X)
3240                                       PRINT "TRY RETYPING IT",
3250                                       INPUT S(X)
3260                                       LET X=X-1
3270                                  CONTINUE
3280                             CONTINUE
3290                        NEXT X
3300                     LOOP WHILE C9=1
3310                     'EJECT
```



```
UPDATE1          11/09/78          12:22:16          page 7

3320              MAT C=ZER
3330              LET X9=1
3340              DO
3350                 LET X=S(X9)
3360                 PERFORM PARAMETER
3370                 LET X9=X9+1
3380              LOOP UNTIL X9>N
3390              PERFORM HEADERS
3400              PERFORM LINE
3410              PRINT
3420              PRINT
3430              DO WHILE MORE #4          'file #4 contains (from Parameter) all the
3440                 INPUT #4: X            'locations of systems that met the criteria
3450                 RESET #1: (X-1)*E1
3460                 RESET #2: (X-1)*E2
3470                 READ #1: D$(1), D$(2)
3480                 READ #2: D(1), D(2),D(3), D(4), D(5), D(6), D(7), D(8), D(9), D(10)
3490                 FOR X2=1 TO N
3500                    SELECT CASE S(X2)    'make the proposed changes to the system
3510                    CASE 1,2
3520                          LET D$(S(X2))=C$(S(X2))
3530                    DEFAULT
3540                          LET D(S(X2)-1)=C(S(X2)-1)
3550                    CONTINUE
3560                 NEXT X2
3570                 PRINT D$(1), TAB(T(1)), D$(2),
3580                 PERFORM DUMP.STATION      'check to see if that is what you want
3590                 PRINT "LOOKS O.K. (YES OR NO)",
3600                 INPUT A2$
3610                 IF A2$="YES"
3620                 THEN RESET #1: (X-1)*E1
3630                       RESET #2: (X-1)*E2
3640                       WRITE #1: D$(1), D$(2)
3650                       WRITE #2: D(1), D(2), D(3), D(4), D(5), D(6), D(7), D(8), D(9), D(10)
3660                 ELSE PRINT "CHANGE NOT MADE"
3670                 CONTINUE
3680                 PRINT
3690                 PRINT
3700              LOOP
3710           CONTINUE
3720           SCRATCH #4
3730           'EJECT
```



```
UPDATE1          11/09/78          12:22:27          page 8

3740        ELSE                                 'code to revise papers
3750           RESET #1: 0
3760           PRINT "NAME, LOCATION OF SYSTEM",
3770           INPUT C$(1),C$(2)
3780           PERFORM HEADERS
3790           PERFORM LINE
3800           PRINT
3810           PERFORM FIND.SYSTEM
3820           IF I1$ <> "Y"
3830           THEN PRINT "NO STATION FOUND WITH YOUR REQUIREMENTS."
3840           ELSE
3850              IF D(1) = 0             'D(1) is a flag&pointer in ---.NUM to indicate papers or not.
3860              THEN PRINT "THERE ARE NO PAPERS ASSOCIATED WITH THAT STATION."
3870              ELSE
3880                 RESET #3         'now we have to find the right paper.
3890                 SCRATCH #5
3900                 LET I1$ = ""
3910                 DO UNTIL I1$ = "Y" OR END #3
3920                    INPUT #3: P2, P1$, P2$, P3$, P4$, P9,
3930                    IF P2 = D(1)
3940                    THEN
3950                          PRINT "AUTHOR:   ", P2$
3960                          PRINT "TITLE:    ", P1$
3970                          PRINT
3980                          PRINT "CORRECT ARTICLE",
3990                          PERFORM ANSWER
4000                          PRINT
4010                          IF I1$ = "N" THEN PERFORM DUMP.3.TO.5
4020                    ELSE
4030                          PERFORM DUMP.3.TO.5
4040                    CONTINUE
4050                 LOOP
4060
4070                 'this code will change an entry in a paper only if the user types in
4080                 'a non null responce to the question.
4090                 IF I1$ <> "Y"
4100                 THEN
4110                       PRINT "ALL PAPERS FOR THAT SYSTEM HAVE BEEN PRODUCED."
4120                       PRINT "NONE FIT YOUR REQUIREMENTS.  PLEASE RECHECK YOUR DATA"
4130                 ELSE
4140                       PERFORM PAPER.INFO
4150                       'EJECT
```

```
UPDATE1          11/09/78          12:22:36          page 9


4160                              IF X2 = 1    'then the abstract did not change
4170                              THEN
4180                                  PERFORM DUMP 3 TO 5
4190                              ELSE        'if the abstract changed, we have to do everything the hard way
4200                                  PRINT #5: P2;",";Z$&P1$&Z$;",";
4210                                  PRINT #5: Z$&P2$&Z$;",";Z$&P3$&Z$;",";
4220                                  PRINT #5: Z$&P4$&Z$;",";(X2-1);
4230                                  FOR X1 = 1 TO (X2-1)
4240                                      PRINT #5: ",";Z$&A$(X1)&Z$;
4250                                  NEXT X1
4260                                  PRINT #5:
4270                                  PRINT
4280                                  IF P9 <> 0 THEN LINPUT #3: P5$
4290                              CONTINUE
4300
4310                              DO UNTIL END #3  'dump the rest of the papers to temp file #5
4320                                 LINPUT #3: P5$
4330                                 PRINT #5: P5$
4340                              LOOP
4350
4360                              SCRATCH #3  'be very careful. There goes the data
4370                              RESET #5    'Scratch file with all the data
4380                              DO UNTIL END #5  'completely copy file #5 into file #3
4390                                 LINPUT #5: T$
4400                                 PRINT #3: T$
4410                              LOOP
4420
4430                      CONTINUE
4440                   CONTINUE
4450                CONTINUE
4460             CONTINUE
4470
4480             PRINT "MORE REVISIONS",
4490             PERFORM ANSWER
4500         LOOP WHILE I1$="Y"
4510 CONTINUE
4520
4530 'EJECT
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
UPDATE1          11/09/78          12:22:45          page 10


4540 DEFINE PARAMETER
4550      IF X <> 12 THEN  PRINT M$(X),           'print header if not paper section
4560
4570      SELECT CASE X
4580
4590      CASE 1,2                                'system name, location data
4600           DO
4610               LINPUT C$(X)
4620               IF LEN(C$(X))>14 THEN PRINT "TOO LONG! TRY AGAIN",    'string can't be longer than 14 characters
4630           LOOP UNTIL LEN(C$(X))<=14
4640
4650      CASE 3,5,6,7,8,10                       'system type, waste type, etc. data
4660           DO
4670               INPUT A$                        'input parameter
4680               LET A1$=SEG$(A$,0,3)            'take first 3 letters
4690               LET X1=VAL(S$(X,1))+1           'number of  choices plus 1 ( skip first S$ since its not a choice)
4700               LET X2=2                        'start with 2nd S$ because of the above
4710               DO
4720                  IF A1$=SEG$(S$(X,X2),0,3) THEN LET C(X-1)=X2-1    'if a hit is scored note which one it was
4730                 LET X2=X2+1                   'up the choice counter by one
4740               LOOP UNTIL X2>X1 OR C(X-1)>0
4750               IF C(X-1)=0                     'if no hits
4760               THEN PRINT  "NO SUCH SPECIFICATION AS: ", A$
4770                    PRINT "THE POSSIBILITIES ARE: "
4780                    FOR X2=2 TO X1
4790                        PRINT "      ", S$(X,X2)
4800                    NEXT X2
4810                    PRINT "TRY RETYPING IT",
4820               CONTINUE
4830           LOOP UNTIL C(X-1)>0                 'loop until a choice has been made
4840
4850
4860      CASE 4,9,11                             'flow rate, app. rate, year data
4870           INPUT C(X-1)
4880           IF X<>11
4890           THEN LET C(X-1)=INT(1000*C(X-1)+.5)/1000       'don't want more than 3 decimal places for these guys
4900           ELSE LET C(X-1)=INT(C(X-1))     'only want whole nos. for years
4910           CONTINUE
4920
4930           'EJECT
```

```
UPDATE1          11/09/78          12:22:53          page 11


4940     CASE 12                               'paper data
4950          LET H=0                          'flag variable to indicate existance of duplicate system (H=1 if so)
4960          'routine to search for a possible duplicate system
4970          LET L1=LOC(#1)                   'save present position in .---.NAM
4980          LET L2=LOC(#2)                   'save present position in .---.NUM
4990          RESET #1:0
5000          RESET #2:0
5010          DO WHILE LOC(#1)<LOF(#1) AND H<>1
5020             READ #1:D1$,D2$
5030             READ #2:D(1),D(2),D(3),D(4),D(5),D(6),D(7),D(8), D(9), D(10)
5040             IF C$(1)=D1$ AND C$(2)=D2$  'test whether system name & location match
5050             THEN LET N2=1                 'counter
5060                DO
5070                .  LET N2=N2+1
5080                LOOP UNTIL N2>10 OR C(N2)<>D(N2)      'test system type, flow rate, etc. for matches
5090                IF N2>10                   'if true then there is a duplicate system already in the files
5100                THEN
5110                     LET H = 1             'set flag for duplicate system found
5120                CONTINUE
5130             CONTINUE
5140          LOOP
5150          RESET #1:L1                      'set .---.NAM to its original position
5160          RESET #2:L2                      'set .---.NUM to its original position
5170
5180          IF H = 1                         'if we found a duplicate system, then error message...
5190          THEN
5200               PRINT
5210               PRINT "DUPLICATE SYSTEM FOUND.  ADDITIONS NOT MADE."
5220               PRINT "PLEASE ENTER PAPERS TO EXISTING SYSTEMS USING THE 'ADD PAPER' SECTION."
5230               PRINT
5240          ELSE                             'if no dupilcate system found, assign a new paper location no.
5250               PRINT M$(X);
5260               INPUT P1
5270               IF P1 > 0
5280               THEN
5290                    PRINT
5300                    PRINT "INFORMATION ON PAPER(S)"
5310                    PRINT
5320                    LET P = P + 1
5330                    LET P2 = P
5340                    PERFORM MANY.PAPER.ADD
5350
5360               ELSE LET P2 = 0             'set location no. to zero -- no papers with this system
5370
5380               CONTINUE
5390          CONTINUE
5400
5410     CONTINUE
5420 DEFEND
5430 'EJECT
```

```
UPDATE1          11/09/78          12:23:02          page 12


5440 DEFINE HEADERS
5450      'subroutine to print headers on terminal
5460      PRINT
5470      PRINT
5480      PRINT "SYSTEM NAME"; TAB(T(1)); "LOCATION"; TAB(T(2)); "SYSTEM";
5490      PRINT TAB(T(3)); "FLOW"; TAB(T(4)); "WASTE TYPE"; TAB(T(5)); "GROUND";
5500      PRINT TAB(T(6)); "TREATMENT"; TAB(T(7)); "APPLIC."; TAB(T(8)); "APPLIC.";
5510      PRINT TAB(T(9)); "PURPOSE"; TAB(T(10)); "DATE"
5520      PRINT TAB(T(2)); "TYPE"; TAB(T(3)); "RATE"; TAB(T(5)); "COVER";
5530      PRINT TAB(T(6)); "TYPE"; TAB(T(7)); "MODE"; TAB(T(8)); "RATE"
5540      PRINT TAB(T(3)); "m.g.d."; TAB(T(8)); "in./wk."
5550 DEFEND HEADERS
5560
5570 DEFINE DUMP.3.TO.5
5580      IF P9 <> 0                           'then there will be more on this line of entry.
5590      THEN                                 'because of the abstract.
5600           LINPUT #3: P5$
5610           PRINT #5: P2;",";Z$&P1$&Z$;",";
5620           PRINT #5: Z$&P2$&Z$;",";Z$&P3$&Z$;",";
5630           PRINT #5: Z$&P4$&Z$;",";P9;",";P5$
5640      ELSE
5650           PRINT #5: P2;",";Z$&P1$&Z$;",";
5660           PRINT #5: Z$&P2$&Z$;",";Z$&P3$&Z$;",";
5670           PRINT #5: Z$&P4$&Z$;",";P9
5680      CONTINUE
5690 DEFEND DUMP.3.TO.5
5700
5710 DEFINE DUMP.STATION
5720      FOR X1=2 TO 10                       'dump out the station legibly
5730           PRINT TAB(T(X1));
5740           SELECT CASE X1
5750           CASE 3,8,10
5760                IF D(X1)<>0                'O signals that the catagory is not applicable or unknown
5770                THEN PRINT D(X1);
5780                ELSE PRINT " N/A ";
5790                CONTINUE
5800           DEFAULT
5810                PRINT S$(X1+1, 1+D(X1));
5820           CONTINUE
5830      NEXT X1
5840      PRINT
5850 DEFEND DUMP.STATION
5860 'EJECT
```

```
UPDATE1           11/09/78          12:23:12          page 13


5870 DEFINE FIND.SYSTEM
5880    LET I1$ = ""                         'reset our question flag for this do-loop
5890    RESET #1: 0                          'reset our name file(.---.NAM)
5900    DO
5910       READ #1: D$(1), D$(2)
5920       IF C$(1) = D$(1)  AND  C$(2) = D$(2)  'did we find a station with the right name?
5930       THEN
5940            RESET #2: ( LOC(#1)/E1 - 1 ) * E2     'check to see if this is the right station
5950            READ #2: D(1),D(2),D(3),D(4),D(5),D(6),D(7),D(8),D(9),D(10)
5960            PRINT D$(1); TAB(T(1)); D$(2);
5970            PERFORM DUMP.STATION
5980            PRINT
5990            PRINT "CORRECT SYSTEM";
6000            DO
6010               PERFORM ANSWER
6020            LOOP UNTIL I1$ = "Y" OR I1$ = "N"
6030            PRINT
6040       CONTINUE
6050    LOOP UNTIL I1$ = "Y" OR LOC(#1) = LOF(#1)
6060 DEFEND FIND.SYSTEM
6070
6080 DEFINE PAP.NUM
6090    IF D(1) = 0                          'when adding paper to system that previously had no papers,
6100    THEN                                 'we must remember to give the system a paper pointer
6110         LET P = P + 1
6120         LET P2 = P
6130         LET P3 = LOC(#2)
6140         RESET #2: P3 - 10
6150         WRITE #2: P2
6160         RESET #2: P3
6170    ELSE
6180         LET P2 = D(1)
6190    CONTINUE
6200 DEFEND PAP.NUM
6210
6220 DEFINE LINE
6230    FOR X1 = 1 TO 133
6240          PRINT "-";
6250    NEXT X1
6260    PRINT
6270 DEFEND LINE
6280
6290 DEFINE QUOTE
6300    PRINT "QUOTES ARE NOT ALLOWED IN TEXT.  USE APOSTROPHIES.  TRY AGAIN"
6310 DEFEND QUOTE
6320
6330 DEFINE ANSWER
6340    DO                                   'make sure we get a yes or no
6350       LINPUT I1$
6360       LET I1$=SEG$(I1$,1,1)
6370       IF I1$ <> "Y"  AND  I1$ <> "N" THEN PRINT "'YES' OR 'NO'.  REENTER:  ";
6380    LOOP UNTIL I1$ = "Y"  OR  I1$ = "N"
6390 DEFEND ANSWER
6400




UPDATE1           11/09/78          12:23:20          page 14

6410 'EJECT
```

```
UPDATE1          11/09/78          12:23:20          page 15


6420 DEFINE PAPER.INFO
6430      'this code gets the information on the papers.  unless a null string
6440      'is input, in which case it leaves the variable unchanged.
6450      '
6460      PRINT "AUTHOR:   ",
6470      DO
6480         LINPUT P$
6490         LET K5 = POS( P$, Z$, 0 )
6500         IF K5 > 0 THEN PERFORM QUOTE
6510      LOOP UNTIL K5 = 0
6520      IF LEN(P$) <> 0 THEN LET P2$ = P$
6530
6540      PRINT "TITLE:    ",
6550      DO
6560         LINPUT P$
6570         LET K5 = POS( P$, Z$, 0 )
6580         IF K5 > 0 THEN PERFORM QUOTE
6590      LOOP UNTIL K5 = 0
6600      IF LEN(P$) <> 0 THEN LET P1$ = P$
6610
6620      PRINT "PUBLICATION:   ";
6630      DO
6640         LINPUT P$
6650         LET K5 = POS( P$, Z$, 0 )
6660         IF K5 > 0 THEN PERFORM QUOTE
6670      LOOP UNTIL K5 = 0
6680      IF LEN(P$) <> 0 THEN LET P3$ = P$
6690
6700      PRINT "PUB. DATE:   ";
6710      DO
6720         LINPUT P$
6730         LET K5 = POS( P$, Z$, 0 )
6740         IF K5 > 0 THEN PERFORM QUOTE
6750      LOOP UNTIL K5 = 0
6760      IF LEN(P$) <> 0 THEN LET P4$ = P$
6770
6780      PRINT "ABSTRACT (HIT RETURN WHEN FINISHED)"
6790      LET X2 = 0
6800      DO
6810         LET X2 = X2 + 1
6820         DO
6830            LINPUT A$(X2)
6840            LET K5 = POS( A$(X2), Z$, 0 )
6850             IF K5 > 0 THEN PERFORM QUOTE
6860         LOOP UNTIL K5=0
6870      LOOP WHILE LEN( A$(X2) ) <> 0 AND X2 < 10
6880 DEFEND PAPER.INFO
6890
6900 'EJECT
```

```
UPDATE1          11/09/78          12:23:31          page 16


6910 DEFINE MANY.PAPER.ADD
6920      PRINT "----------"
6930      FOR X1 = 1 TO P1
6940           LET P1$ = P2$ = P3$ = P4$ = ""
6950           PRINT
6960           PRINT "#"; X1
6970           PRINT
6980           PERFORM PAPER.INFO
6990           PRINT #3:P2; ","; Z$&P1$&Z$; ","; Z$&P2$&Z$; ","; Z$&P3$&Z$; ","; Z$&P4$&Z$; ","; (X2-1);
7000           FOR X4=1 TO (X2-1)
7010                PRINT #3: ","; Z$&A$(X4)&Z$;
7020           NEXT X4
7030           PRINT #3:                          'in effect, a carriage return
7040           PRINT "----------"
7050      NEXT X1
7060 DEFEND MANY.PAPER.ADD
7070
7080 END

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

TERMINUS          UPDATE1

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
12:23:42                                                                                                           11/09/78
```